SATYA VDAPALLI

1126828

Load balancing in distributed query management at web enabled systems

MSc Computer Science
Masters Thesis Report
Department of Computer Science & Technology

Dr WEI HUANG

2012/13

University of Bedfordshire

Thesis author consent form

AUTHOR'S NAME: SATYA VADAPALLI

TITLE OF THESIS: LOAD BALANCING IN DISTRIBUTED QUERY MANAGEMENT AT WEB ENABLED SYSTEMS

DEGREE: MSC COMPUTER SCIENCE

Please read carefully and sign the following as appropriate.

I have read and understood the University's regulations and procedures concerning the submission of my thesis.
I understand that I have already signed a declaration agreeing to my dissertations being kept in the Learning Resources Centre (LRC) when I enrolled.
We would like now, to extend this agreement by making the thesis available online.
Further to this,
I AGREE AS FOLLOWS:
- That I am the author of the work.
- That I have exercised reasonable care to ensure that the Work is original, and does not to the best of my knowledge break any UK law or infringe any third party's copyright or other Intellectual Property Right.
- The LRC and BREO administrators do not hold any obligation to take legal action on behalf of the Depositor (you), or other rights holders, in the event of breach of intellectual property rights, or any other right, in the material deposited.

DELETE ONE OF THE FOLLOWING OPTIONS AS APPROPRIATE:

1. I hereby extend my consent to this thesis being included in the LRC as well as on BREO via online access.
2. I hereby DO NOT extend my consent to this thesis being included in the LRC as well as on BREO via online access.

AUTHOR'S PERSONAL SIGNATURE: SATYA VADAPALLI

AUTHOR'S STUDENT NUMBER: 1126828

DATE: 24/05/2013

Abstract

Every day more and more Business are using web as compulsory medium to provide services. Ever increase in technological advancement leads more devices and applications are accessing web based application around the clock. So single node web applications are prone collapse, designing an application with Distributed frame work and mange those application often reduce the risk of single point failure. In that strategy load balancers plays an important role to direct the traffic to multiple Servers. Existing load balancers are Prone to fail and centralized strategy to distribute the load among the server. Our proposed heuristic based load balancer follows Decentralized approach to solve the problem and our ANN based supervised back propagation technique gives optimized results that existing Load balancers.

Acknowledgement

This is the time to acknowledge my supervisor Dr WEI HUANG li for his support throughout the thesis work and my parents for their support to fulfil my Masters.

Dedication

This work dedicated to my parents

Satya Vadapalli

LIST OF ABBREVIATIONS

DCE - DISTRIBUTED COMPUTING ENVIRONMENT

DFS - DISTRIBUTED FILE SYSTEM

COBRA - COMMON OBJECT REQUEST BROKER ARCHITECTURE

RM-ODP - REFERENCE MODEL OF OPEN DISTRIBUTED PROCESSING

ANN - ARTIFICIAL NEURAL NETWORKS

LIST OF FIGURES

LIST OF TABLES

Table of contents:

Chapter 1
Introduction

1.0 Introduction

In recent decade usage of technology in our day to day lives ever increased and reaches to maximum extent. Now increase in mobile technology data consumption and creation was huge. The existing technology to power internet been always transforming its shape and size to cope up that demand. Every web based application was typically powered by client and server model. Clients are typically web browsers, specially designed application programming interfaces (API) and mobile hand held devices with apps installed in them. Servers contain webserver to serve web pages and database servers to store user data.

Present trend creates pita bytes of data in a few years' time .so the way to organize such huge amount data is challenge for existing client and server technology , design clusters for such data was incurs more cost and complexity

1.1 Problem Definition

Ever increase in web based application often leads to load problems and application scaling issues at Servers. Existing load balancing techniques does not having full control of the application at Clients browser end. Proposed load balancing technique based on artificial neural networks (ANN) model system design able to control the load among the servers with dynamic control of application load among the servers with clients weights(minimum cost) considered.

1.2 Aim

Design a load balancer with dynamically adjust load among the servers in Distributed web based systems. The proposed Heuristic based load balancer able to train itself for the changing requirements of ever changing web systems.

1.3 Objectives

The main objectives are

- Proposed Load balancer is dynamically adjust the clients requests among the servers.
- Takes clients side weight into consideration.
- Effectively manage the system and network resources.
- Reduce query processing time.
- Increase overall efficiency of the web based applications

1.4 Methodologies Used

They are many methodologies available to design, develop and deploy software systems. i.e waterfall,spiral,agile. For our proposed system I used agile methodologies for analysis, design and development phases.

1.5 Proposed Thesis organization

This Thesis document was organised in the form of chapters and sub divides in to sections.

Chapter 1: This chapter was compromised with introduction about the proposed Thesis, followed by problem definition, aim and objectives of proposed thesis work, available methodologies and thesis organization.

Chapter 2: This chapter was compromised with contextual review, how existing technologies are used to design load balancers, existing algorithms and its application in design of load balancer, artificial neural networks (ANN), and use of back propagation techniques. Advantages of existing load balancers. Disadvantages of existing load balancers.

Chapter 3: This chapter was compromised with proposed system design considerations, how the overall system was designed in agile methodology with help of data flow diagram (DFD).

Chapter 4: This chapter was compromised with proposed system development, with following agile methodology proposed system artefact was analysed and developed in phase's .This phases was sub divided in to Modules. Each Module was developed as per the system requirement.

Chapter 5: This chapter was compromised with proposed system information about deployment and testing .This deployment was tested with proposed consideration and its methodologies. In testing I used

manual testing approach, white box testing for code review and black box testing for graphical interface.

Chapter 6: This chapter was compromised with proposed thesis conclusion and further enhancements can be made to proposed system.

Chapter7: This chapter was compromised with Reference's, Interim report and poster.

University of Bedfordshire

Chapter 2
Contextual Review

2.0 Contextual review

In past we have legacy systems where one computer servers one user at a time ,then time passes one computer server contains many applications and servers number of users, later user numbers are increased ,applications also increased we keep on increasing processing power to server the user requests. Then we have new computing paradigm called cluster computing where one problem set was solved by group of tightly integrated computers, then we have grid computing where loosely integrated computers solved problem sets. Now we are in cloud computing era where all computing power resides in geographically distinct location and accessing from thin clients.

In distributed computing data resides in more than one node, the query which access those data from different nodes needs to rewrite to get optimal execution plan. At the same way the request coming from different clients' needs to send to different nodes to address load balancing on the server systems.

Ever increase in number of clients accessing the server systems; load balancing is an important task to facilitate the optimal resources utilization. Without load balancers some servers are keep on busy and some servers are always free and some client's requests are always in starvation mode. To address all this issues load balancers are resides in between server systems and clients. These load balancing in designed in two counter parts one is looking at server parameters and another is at client requests. And these load balancers are broadly divided in to two types

2.1 Types of load balancers

2.1.0 Hardware load balancers

These types of load balancers are designed specifically with as per hardware and application requirements and are very expensive and give better performance.

2.1.1 Software load balancers

In software load balancer are specially programed software to intelligently handled request of clients and passed to the servers to process that requests. This gives less performance than hardware load balancers and gives optimal performance and provides greater usability.[1,2,3]

2.2 Load balancing algorithms

2.2.0 First come first server (FCFS)

In first come first server (FCFS) algorithm all input requests are placed in a queue and then server which ever next available server to server the request. In this type of algorithm some servers are always busy or some might always free.[1,2,3,6]

2.2.1 Shortest remaining processing time (SRPT)

In Shortest remaining processing time algorithm the scheduler first estimate the job execution time and then send which have less execution time first ,in this case jobs whose job have longest execution time must me waited along or may be in starvation mode. If all jobs have equivalent processing time it gives good performance than FCFS[2,6]

2.2.2 Process sharing

In this type of algorithms, server process are all clubbed together then shared among the requests, irrespective of job execution time .and this type of algorithms are easy to implement but less adaptability because nature of application is important before adopt this algorithms.[2,6]

2.2.3 Round Robin (RB)

In round robin technique all servers are in circular fashion each request is passed to servers one at a time to server example consider four servers and assigned names SER1,SER2,SER3 AND SER4. The requests are allotted in this manner REQ1 to SER1, REQ2 to SER2, REQ3 toSER3, REQ4 to SER4, REQ5 to SER1.[2,6]

2.2.4 Multiple Queue scheduling

In Multiple Queue scheduling algorithms jobs or requests are first segregated as job type and then passed to different set of queues and then adopt each and every queue with different scheduling algorithms some might adopt round robin others are First come fist server. Generally high priority jobs such as system process are given first priority with RB scheduling algorithms.[2,6]

2.2.5 Weighted round robin

In weighted round robin technique servers are classified based on its processing capabilities then assign those with unique weight value then pass jobs to servers in a queue then apply round robin technique for the assigned jobs in particular weighted queue.[2,6,21]

2.3 Artificial intelligences technique

Load balancing at distributed dynamic environment often implemented at different levels depend upon type of the application architecture. In today's web environment often uses multitier environment where each and every cross section we have load balancers to adjust the load and gives optimize usage of resources. Example of load balancers are DNS load balancers, cluster load balancers, database load balancers, web server load balancers, cache server load balancers.

Development Artificial intelligences gives a new approach to handle complex tasks in heuristic way and it provides many techniques to solve vast amount of problem sets.

2.3.0 Artificial neural networks (ANN)

In artificial neural networks systems was designed in form of human brain. This consists of neurons. Same concept was implemented in artificial neural networks. This type of systems consists of input neurons ,computational neurons and output neurons ,each neuron associated with weight ,and internal neurons are inter connected with other neurons , the information flow between neurons was defined with volume the weight and direction of the information flow assigned with the neuron will take high precedence. The sum of all neurons goes to transfer function and then goes to activation function in such a way neural networks are trained for the given task.

The following figure gives the typical ANN design

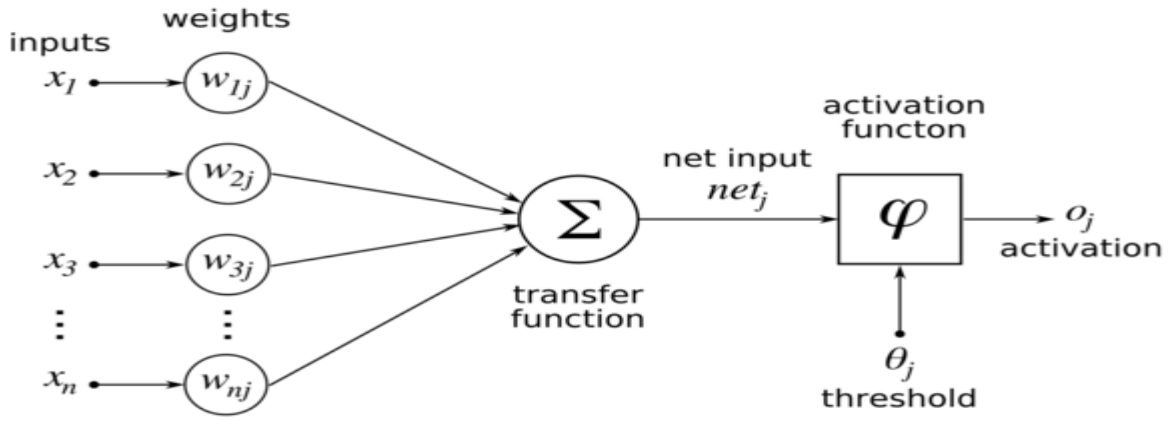


Figure-2.3.0(A):Back propagation

2.3.1 Back propagation Technique:

In back propagation technique we can train a neural network based on required output with an activation function we can train network for any function. Its comes under supervised learning type.

This approach all input weight are summed and passed to transfer function check with activation function. The resulted value was compared with output value calculate the error rate and pass the value back to hidden networks. The process calculates again with changed value until it satisfies the desired output. The ANN training was depending upon the learning rate.

2.4 Drawbacks of existing load balancers:

Existing load balancers are designed for to optimize the load on the servers, which are rely on the server parameters .if one server fails load on the system was distributed to the evenly to other servers but query time increased and once the clients quires are forwarded to servers it never be adjusted and often needs to resend, because TTL (time to live) error are occurred, which lead jobs under starvation.

2.5 Distributed application frame Work:

They are many frame work available to design a distributed application that are mainly "distributed computing environment(DCE)","distributed file system(DFS)"," common object request broker architecture(COBRA)","reference model of open distributed processing(RM-OTP)".[21,22]

University of Bedfordshire

Chapter 3
System Design

3.0 System design methodology

In order to design the proposed load balancer for the large scale distributed web systems, I followed a new novel approach from the artificial neural networks (ANN). In artificial neural networks, there is supervised learning algorithm called back propagation technique. From the literature we have many load scheduling algorithms are available. I followed combination of weighted round robin technique and back propagation technique to design the proposed System design. This proposed methodology gives optimized load balancing strategy for ever changing system load.

3.1 Design of large scale web based systems with load balancing support and multiple parallel query serving

large scale web based applications are designed for 24X7 usage , This complex systems are multi tyre systems which are mainly divided in to three tires that are database tire to hold large amount data ,web tier contains web server ,it servers the web pages and form server it generates forms on fly as per the clients requirements. This complete system was span across multiple physical systems to form cluster or a distributed computing environment. In order to balance the load among the physical systems. Applications are designed to use parallel query processing across the multiple tires.

3.2 Application of ANN in our methodology

In order to design a load balancer for our proposed system I used back propagation technique from artificial neural networks supervised learning algorithms. In this technique I mapped out nodes are our client requests service time, input nodes are query execution time in server, available

free slots for requests, client's location weights . I used weighted round robin technique in scheduling for service in clients.

3.3 System design consideration:

In order to design our proposed load balancer I considered following requirements, designing clients which works with server applications with load balancer enable support , design of server application able to work with load balancer ,server is able send its status to load balancer for given specified time intervals.

3.3.0 Design of ANN based load balancer with the help of proxy's servers:

Proposed load balancer was designed with the help of proxy servers, load balancer was able to receive the requests from the clients then take weight values from clients, takes number of available slots from the server systems , processing times from the servers , number of available free resources from the servers ,take these are all inputs from servers and assign the values to inputs nodes of ANN network , takes input weights from web clients system such as use cookies to identify the system location, take that value and map to assigned value in input weight of a computational neuron. Assign each input query a unique number and send to circular queue ,in circular queue they are some weighted queues each and every query is assigned to pre-defined server identified by load balancer such that all client requests are served with minimum turnaround time. If any of the server down after assigning web client requests, load balancer reassign the values to remaining Server systems. For clients systems see as an proxy servers. So clients are never known that they are communicating to proxy server.

3.3.1 Design of clients with weight based systems:

Our proposed load balancer need special kind of support cookies from web clients. Which contains information related to location of web clients so that load balancer first checks whether any nearest server available to service its requests if not it check for next nearest available server Such a way all clients request are served with optimal or minimum resources used. And these cookies always update itself its average service time, a unique number from service done by the server, next request is served without any further computational calculations.

3.3.2 Design of Servers with load balancer enabled support:

To design a proposed server first consider what are the parameters required for load balancers then pass that values to a load balancers nodes. That variables include number of available slots in each server and server sends report to load balancer informing that what type of queries it can serve fast by knowing list of available parsed statements in cache. These list is passed to different nodes in load balancer then load balancer updates corresponding proxy servers so that they proxy's Update their information is always update in order to change its setting on dynamic environments. Server updates its information in server modules so server module is always asses its information before sending to load balancers. Then Load balancer update its corresponding settings in all client modules and server modules.

Chapter 4
Development of proposed system

4.0 Development of proposed system

The development of proposed system i.e load balancer for distributed web applications system. To design the proposed system I followed agile software development methodology .The complete system was developed in phases that are analysis, design, develop, test and deploy the developed system.

4.0.1 Requirement analysis

To design the proposed system first gathers the required input and proposed algorithm and expected output. Once gather input requirements .then divide those requirements in to two categories that are software requirements and hardware requirements

4.0.1.1 Software requirements

To develop the proposed scenario I choose java

- Java development kit (net beans IDE)
- Mysql server
- Virtual environment(VMware, Virtual Box)

4.0.1.2 Hardware requirements

To develop the proposed scenario it needs minimum hardware requirements.

- 1.6 ghz processor
- 512 mega bytes of random access memory
- 1000MB of storage to store configuration file
- Local area network to test scenario

4.1 Overall system architecture

The complete proposed system was divides into three phases first one design and develop load balancer, second one is design and development of servers and third phase was design of clients to access the server system. proposed system consists of server this server includes web server to service the client pages ,database servers to store the data of clients in form binary objects or normal RDBMS data, may contain forms server to service requested forms .

Normally once the client machine make a request to web based application that request is first received at server end then passed to load balancer some time called as dispatcher , this load balancer was designed to calculate the optimized processing time and distribute the requested queries to different servers then assign this load to one proxy server. That proxy server is responsible to provide the requested response to clients. These clients are associates with use of cookies these cookies collect information such as client location based on ip address of the client. Based on that load balance calculate and update proxy server to service the clients requests and its calculated queries. The following diagram represents overall system architecture

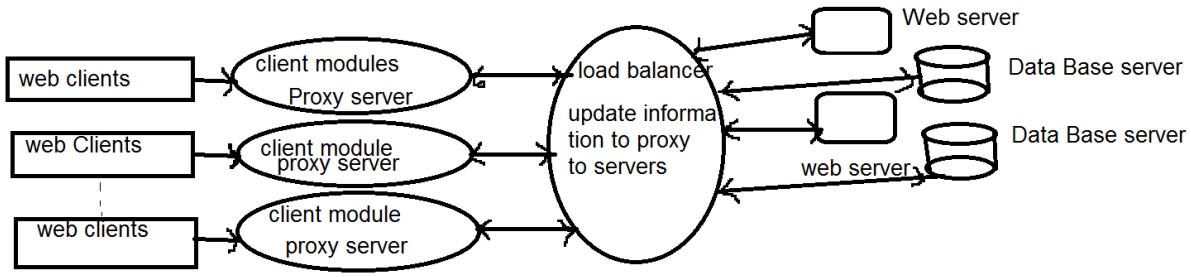


Fig 4.1(A): Proposed overall system design

From system design and requirements analysis I used agile software development methodology to develop this system and divides into list modules

- Development of load balancer
- Development of servers
- Development of client system

4.1.0 Development of Load balancer

Development of load balancer was mainly divided into sub modules that are collecting parameters from server, collecting client's requirements apply proposed logic to calculate optimized execution plan minimum use of resources.

First clients request some service from the web application ,that request is logged into the server side module this module update the request to load balancer , in load balancer waiting queue each request is assigned by unique id and all requests are passed to computational phase in computation phase it calculate the nearest and optimized execution plan for the clients are update the one proxy server to service its request.

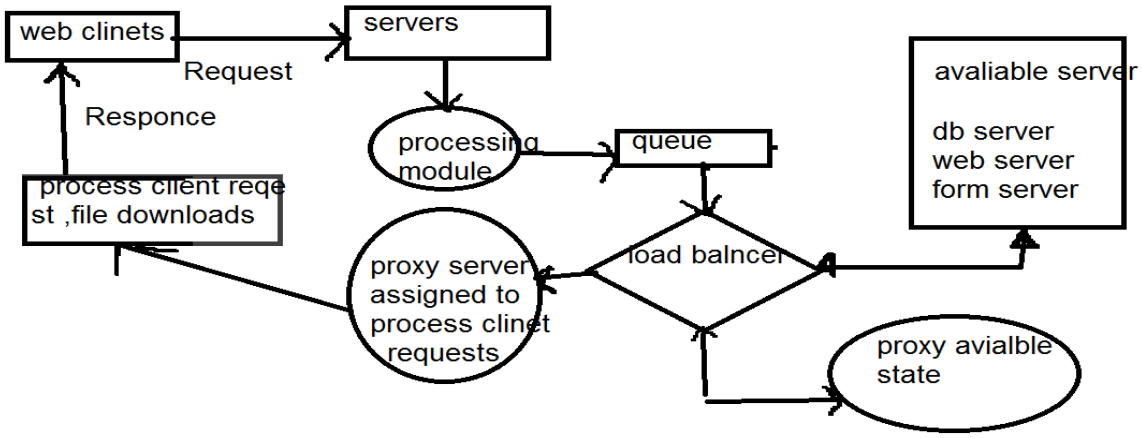


Fig-4.1.0(A): DFD of load balancer

I used java hibernate and swings for graphical interface to design front end of proxy servers. The below figure represents proxy load balncers with proxy server interface.

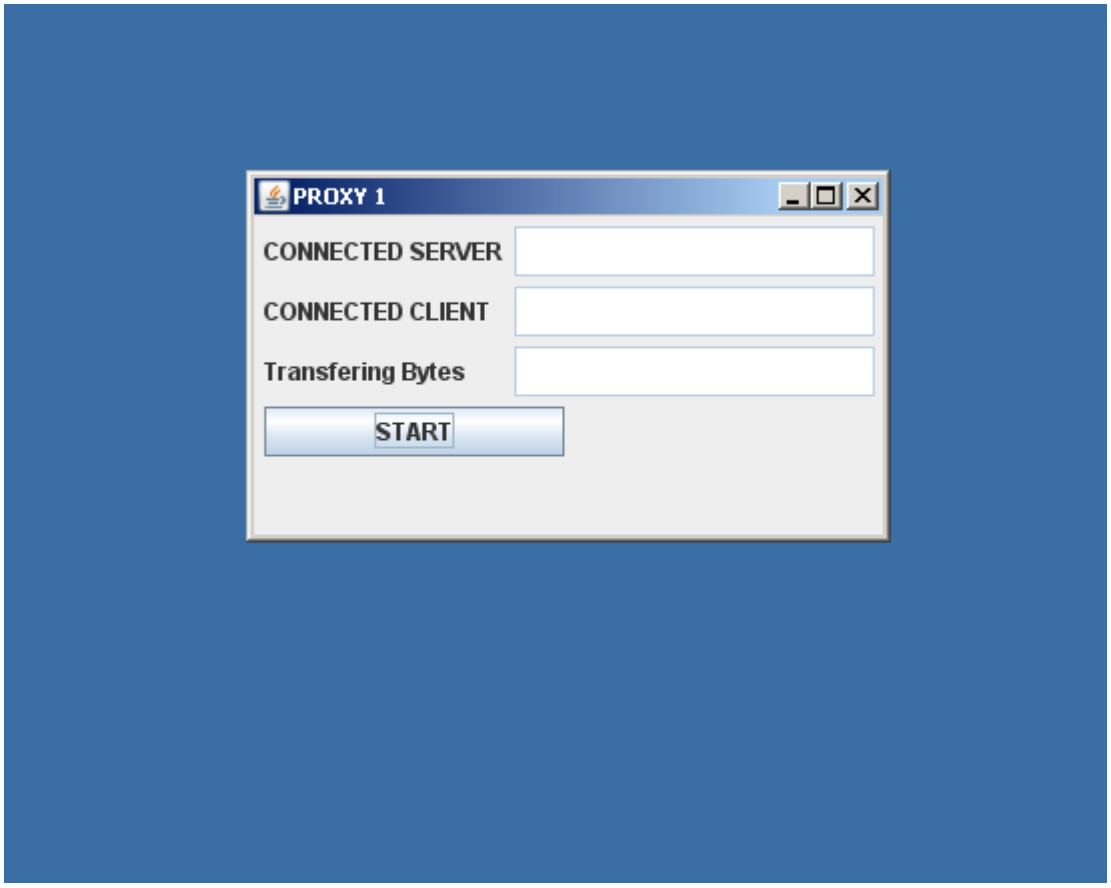


FIG-4.1.0(B): load balancer proxy server interface

4.1.1 Development of Server modules

To develop the server module for the proposed system, I considered a server which contains some files, these files able to download from the client machine. To develop such scenario I designed simple graphical interface in java for server. Like this type of servers is simulated for real world scenario. Once theses server are ready first start servers in network environment. Server contains load balancer module it always collect the system statistics filer those statistics as per load balancer requirements then update timely basis in such way that these servers

are communicating with the load balancers in form of proxy server. The following data flow diagram represents server module

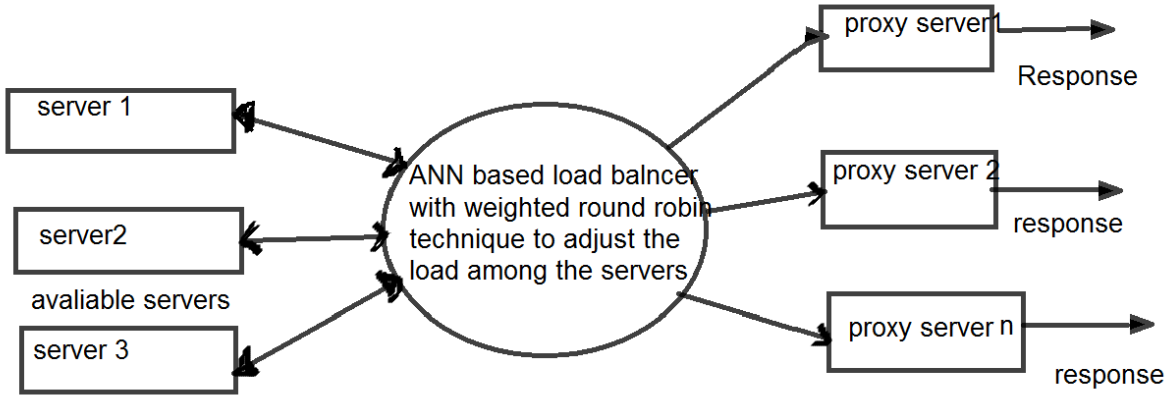


FIG-4.1.1(A): Data flow diagram of server module

The below diagram presents server design graphical interface

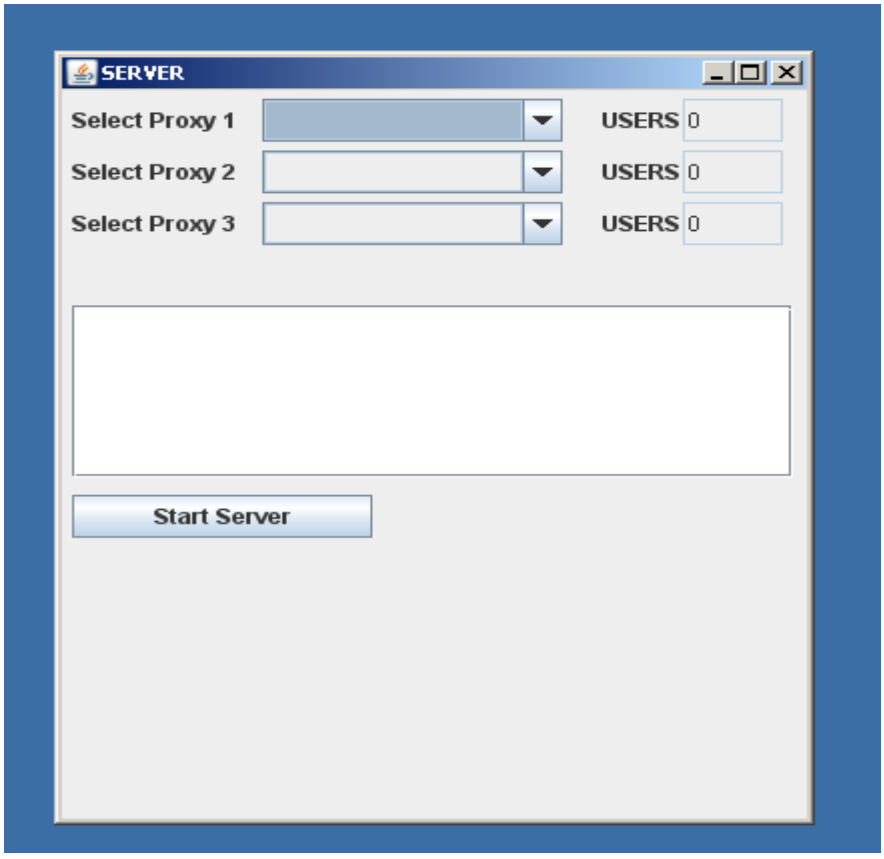


Fig-4.1.1(B): Server GUI

4.1.2 Development of Client modules

As per our proposed system client requirements module all clients are able to send requests to large scale web application .then server receives that request and give response as per the application logic ,I designed a sample java based clients to download files from servers to local machine .these request is being served by servers in our local area network (LAN) environment . The proposed requirements are satisfied in our developed client system.

Following figure data flow diagram of proposed client with download request.

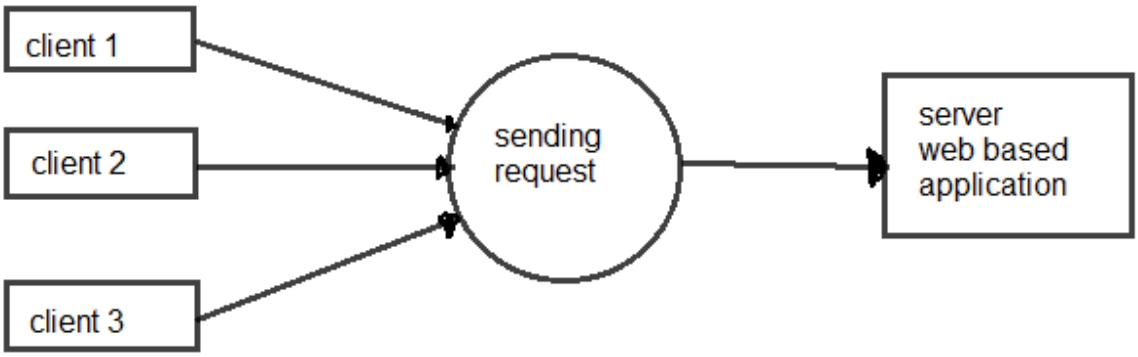


FIG-4.1.2(A): data flow diagram of client systems

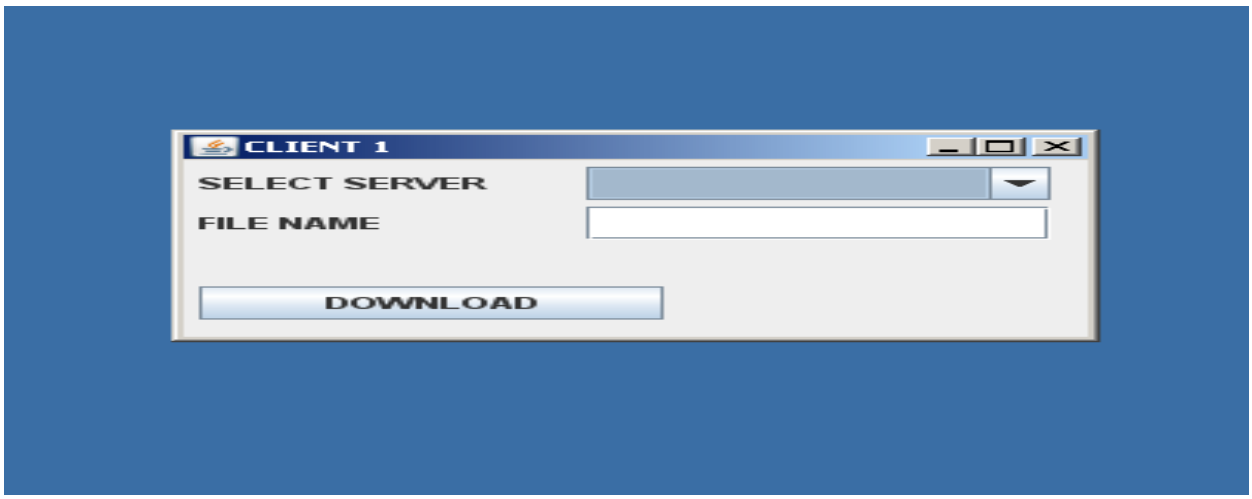


Fig-4.1.2(B): client GUI

4.2 Integration of complete system

Once all modules development is completed we need to integrate all modules in to single workable system. So that I start with server it then integrated with load balancers then check both configurations are correct then go forward integrate with single client then simulate complete system in a single distributed environment in local area network with different systems. Once successful integration check sample data then to testing phase as per agile software development methodology.

University of Bedfordshire

Chapter 5
Testing and Deployment of proposed system

5.0 Deployment of proposed system

The developed system was deployed in virtually created network environment with “windows xp “operating systems, these systems are connected in internal LAN segment. So in that LAN segment each and every system is assigned by unique IPv4 address with hostname associate with it our designed server is able it identify all the connected system host names. Then our load balancer starts and calculates nearest one with minimum load, assign that request to nearest proxy server. So that assigned proxy servers distribute the load among the available servers. Our designed system able run in single virtual host machine the following figure represents running designed system in a single host

5.0.1 Starting server

Our designed server contains some files available for client systems to download. The following figures shows before start server and after start server with load balancer enables .

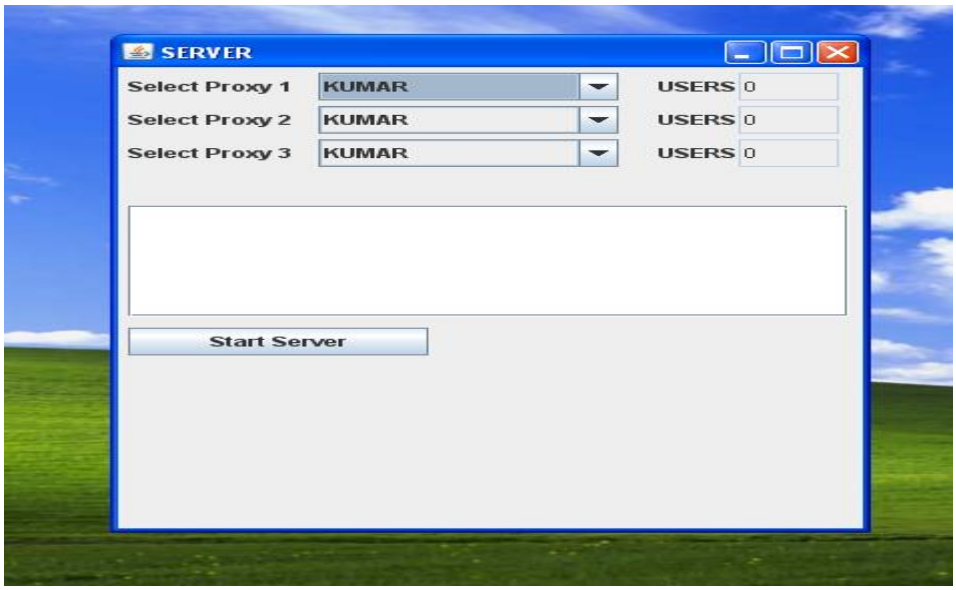


Figure-5.0.1(A): server

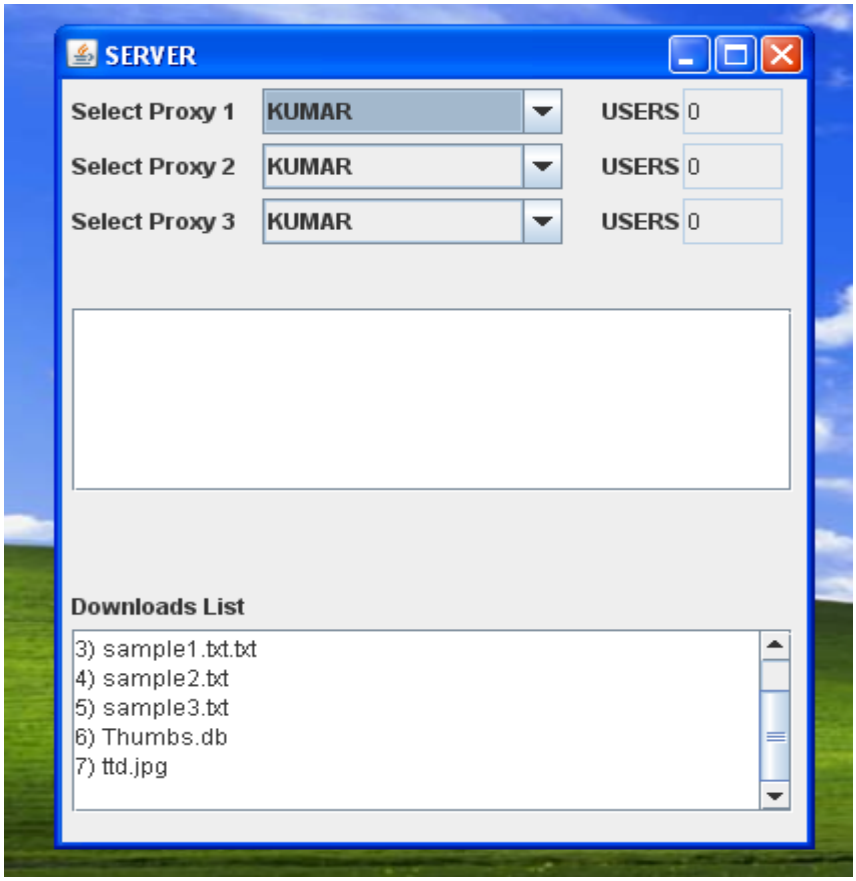


Figure-5.0.1(B): server with enabled load balancer

5.0.2 Starting load balancers

Our developed load balancer works with server modules then they act independently even one of the server or load balancer or any node in between fails they cope with that failures. Below figure shows load balancers at different nodes.

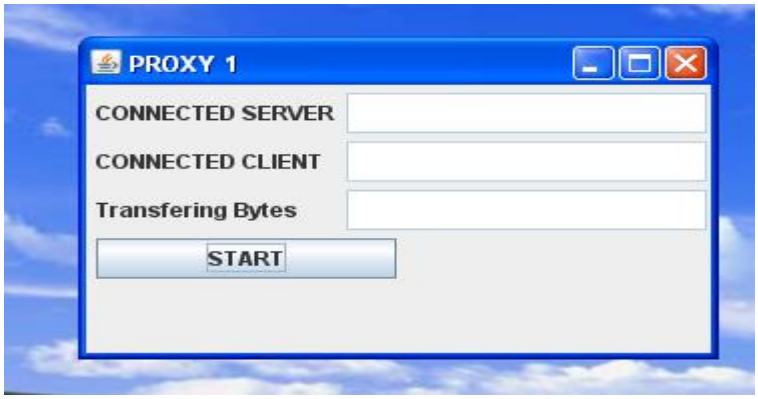


Figure-5.0.2(A): Load balancer

5.0.3 Starting clients

Our developed clients systems able to send its information to load balancer and then ask a service from the server , in server they are many services available our developed scenario is files available to download from it. Given figure represents client asking file to download.

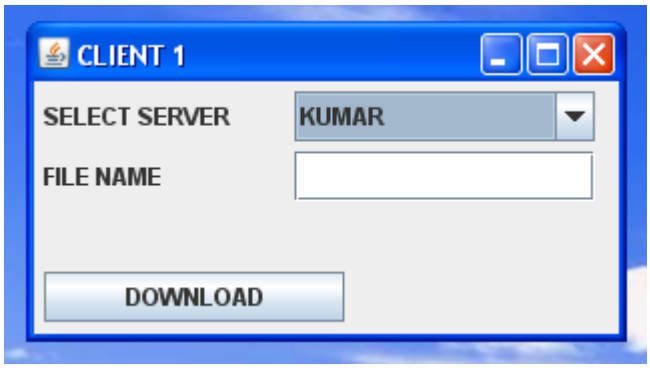


Figure-5.0.3(A):client

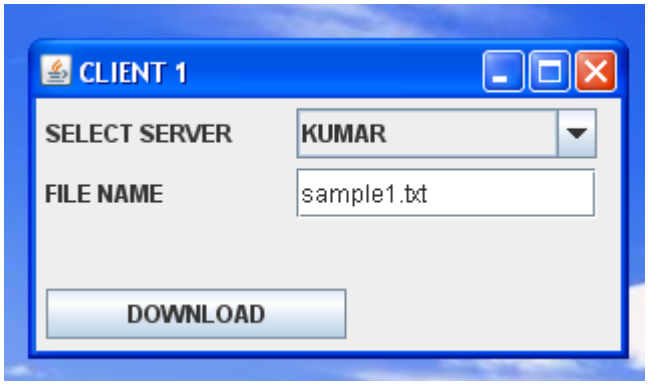


Figure-5.0.3(B): clients requesting a file

5.1 Testing designed system

Testing of prosed design of a system and its development was performd by using manual testing approach used both “black box testing”[] for “Graphical user interfaces(GUI)”[] and internal logic of algorithms

implementation I followed “white box testing” The following tables represents how the tested are performed on developed system.

Server Testing	
Start button	Button displayed
Display file names in server field	Successfully displayed
load balancers 1(proxy1)	Drop down list to display servers
load balancers 1(proxy2)	Drop down list to display servers
load balancers 1(proxy3)	Drop down list to display servers
Display user field	Shows number connected users

Table-5.1.(A): Server Testing

Load balancer	
server	Display which server connected
Clients	Display which Client connected
File transfer	Display file transfer information

Table-5.1(B): Load balancer testing

Client systems	
Select server	Select server name
Select file name to download	Entered file download
Download Button press call	Initiate download file

Table-5.1(C): Clients systems GUI

5.2 Testing of proposed load balancer working

In order to test the proposed novel approach for load balancer in web based distributed systems, I developed an sample artefact to test the case scenario. Which contains distributed file servers where files are located in multiple servers? Java based clients are request a single or set of files from server , in that request load balancer adjust the load among the servers in the form of proxy servers and then start serving clients individual requests process downloads from the nearest available server .

The following figures represent series of tests performed to check our proposed load balancing methodology. Start the servers in distributed

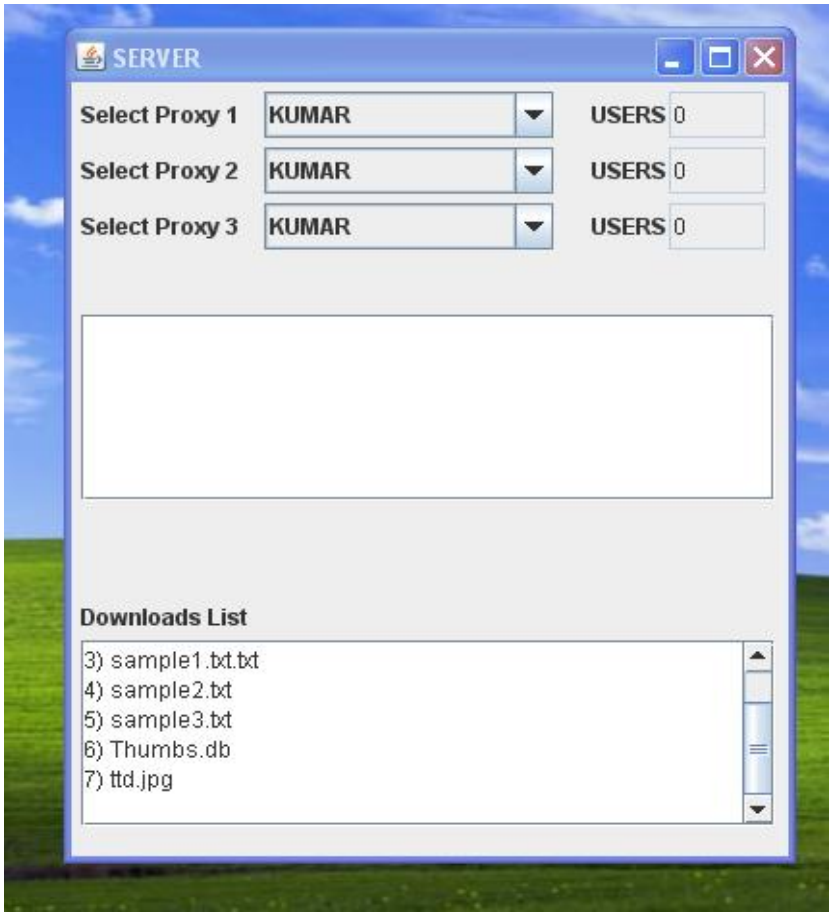


Figure-5.2(A): Server at Kumar system

Start load balancer at “kumar” system with three independent modules called proxies balancers.

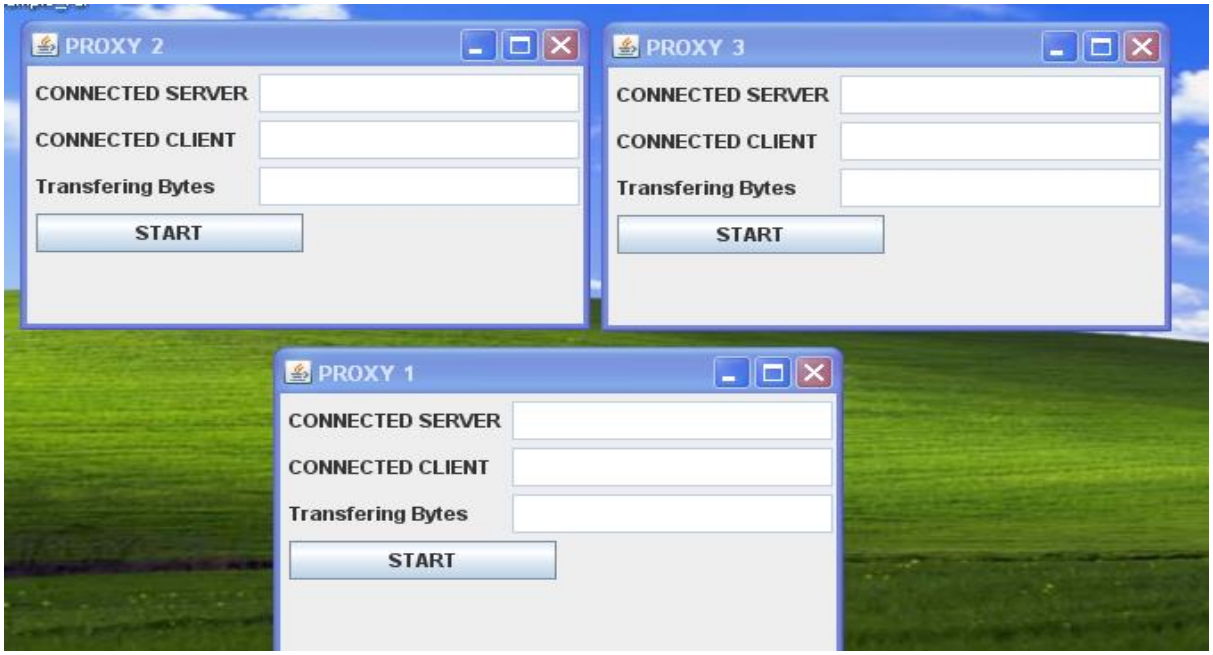


Figure-5.2(B) : Load balancer without start

The start load balancers with name of server that load balancer works on it by choosing server system name at server side and start button

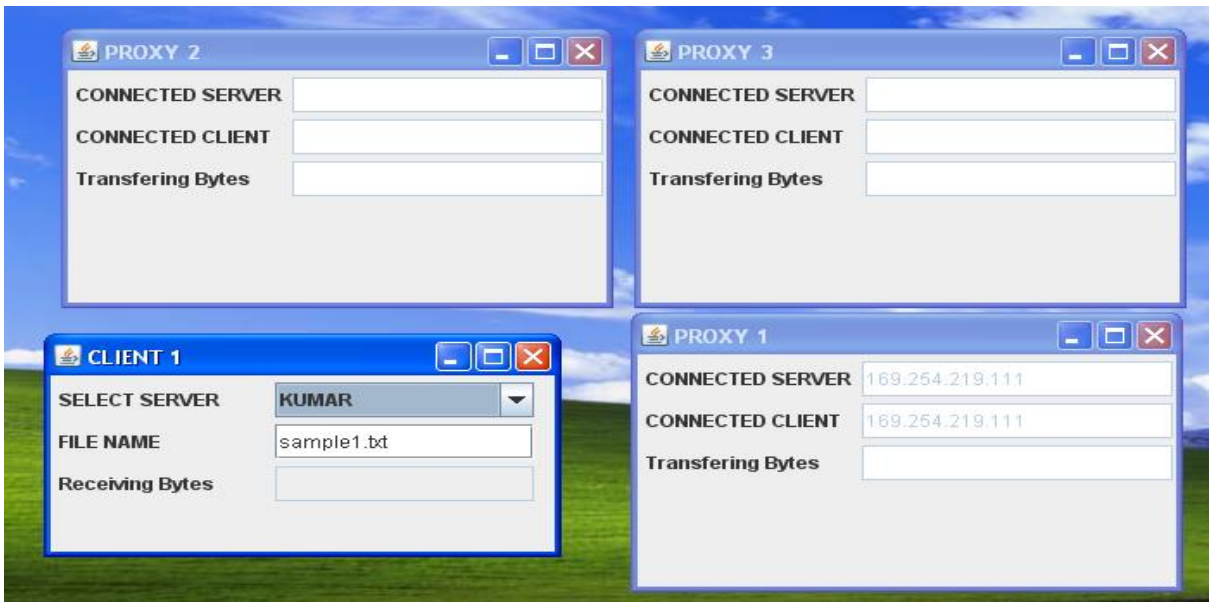


Figure-5.2(C): Shows started Load balancer

Start the client system make a file download request from the server.

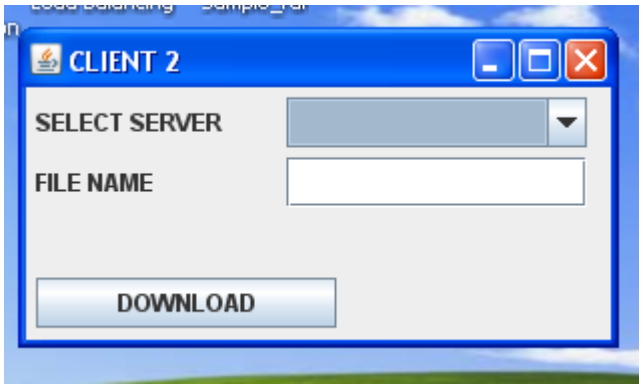


Figure-5.2(D): client system ready to connect

Once client system ask a file from server give file name and server name from which file is downloading.

Our case client system1 and client system2 asking sample 1 to download.

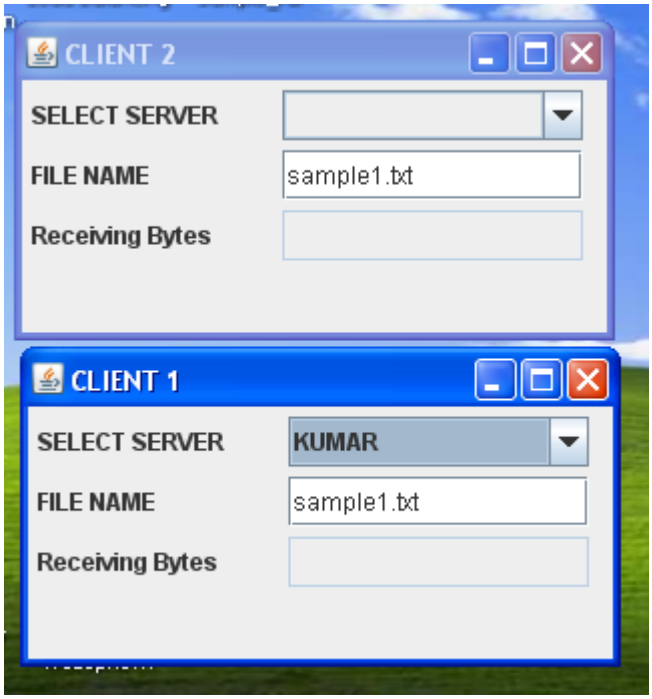


Figure-5.2(E): requesting files from server by clients.

When the clients systems requests a service from servers corresponding load balancer representing that's requests by requstee(clients) and requester(server) status are displayed by load balancer to user.

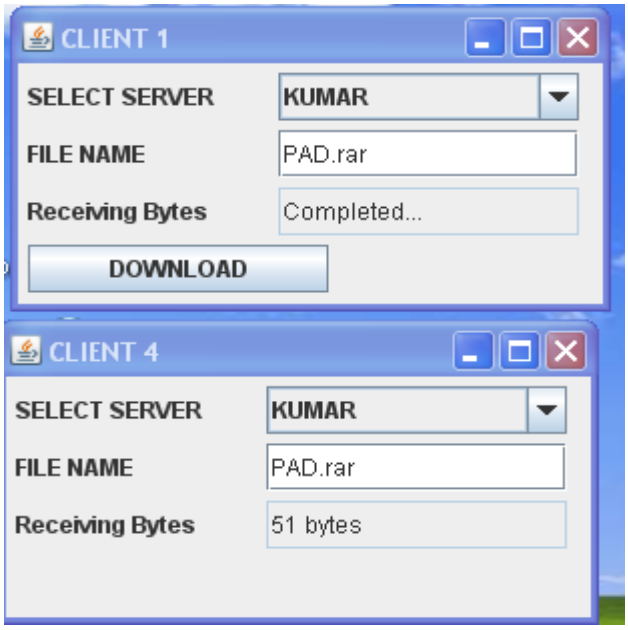


Figure-5.2(F): Clients requested PAD.rar Download

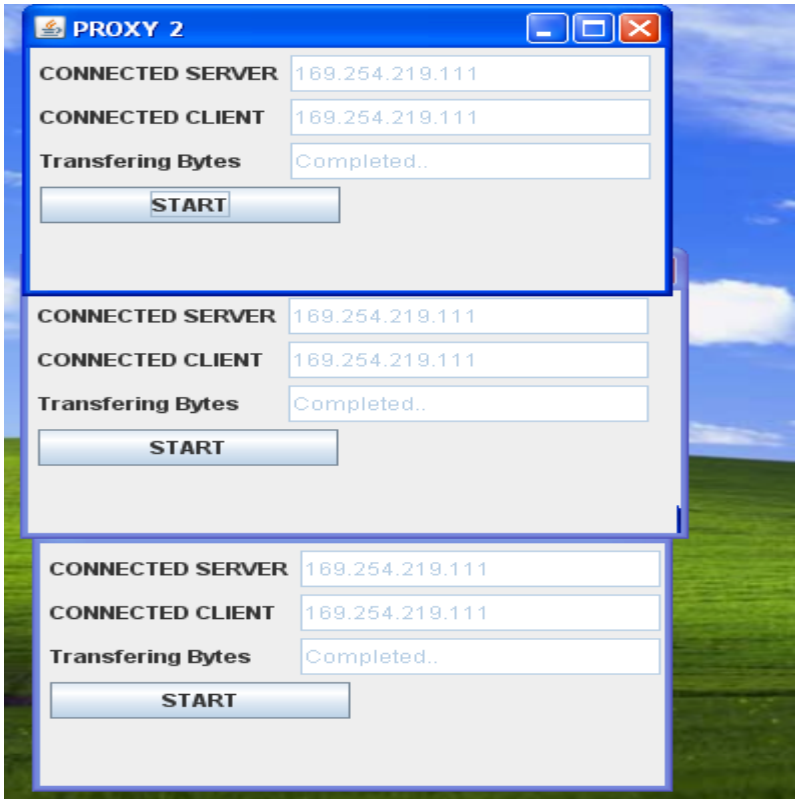


Figure-5.2(G): Load balancer working

In each server shows which users are connected ,which load balancer servers requests its service and available list of service are ready to provide is displayed.

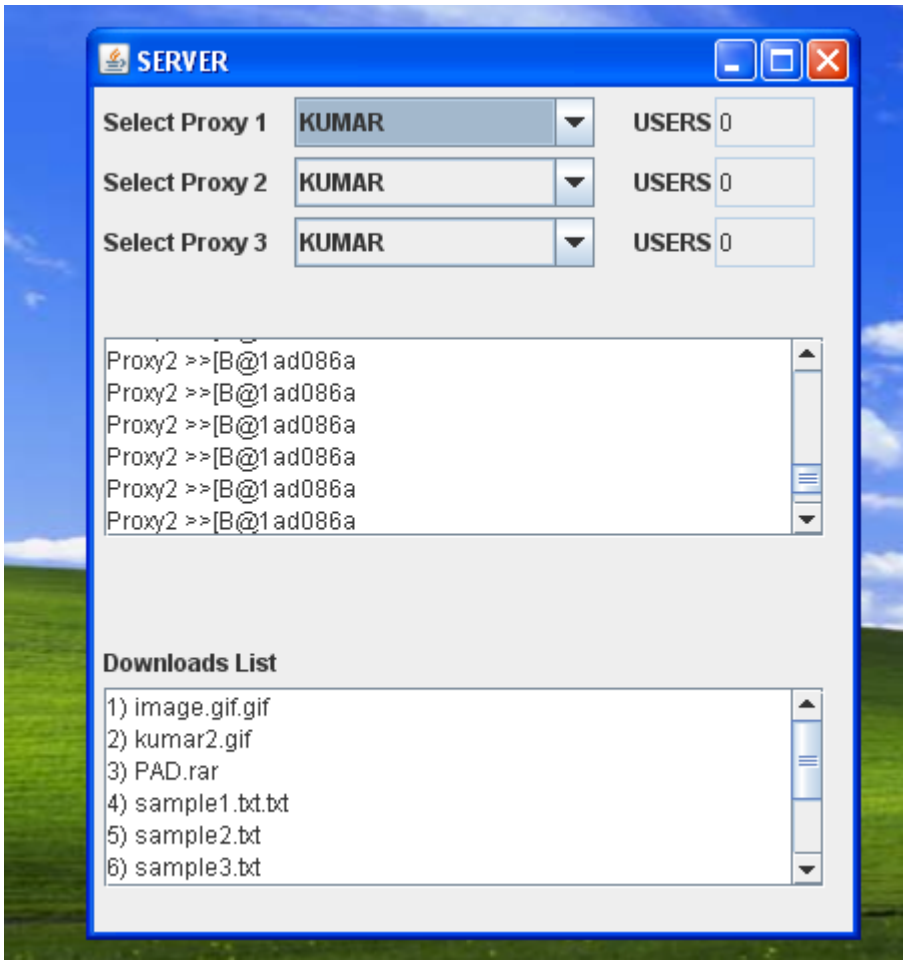


Figure-5.2(F): Server Information to User.

After success full deployment of proposed load balancer satisfied our design requirements and shows expected results, hence this proposed methodology works in distributed web based systems.

Chapter 6
Conclusion

6.0 Conclusion

As we discussed existing methodologies, designs and its corresponding algorithms for Load balancers in our contextual review. Each have own advantages and disadvantages of each other. Our proposed heuristic Load balancer for Distributed query management for web enabled systems shows better results than other load balancers. i used decentralized approach to design a load balancer with a supervised learning technique such as back propagation method from artificial neural networks . So this approach provides dynamic adjust of load among the connected systems, its weight based criteria enables us to consider clients system parameters such as location based service request, available bandwidth at client's networks by using existing web based cookies method.

This proposed system design provides decentralized control of resources with dynamically changing resources consumption.

6.1 Limitation Of proposed load balancer:

While designing load balancer I used artificial neural networks supervised learning technique Back propagation. This technique is slower than some existing load balancer algorithms.

6.2 Future enhancements for the proposed system

- Failure of any server in the system is automatically transfer running queries to rest of the server with combination of Virtual monitoring agents such as "Virtual network interface "
- Auto regeneration of proxy servers to adjust the load in case of failure in some Node failures.

Chapter 7
References

7.0 References:

[1] Donald kossmann “the state of art in distributed query processing” published in ACM computing serveys,vol.32 no.4 December 2000,pp 422-469.

[2] Peter Membrey,David Hows,Eelco Plugge “ Practical load balancing” book by Apress -2012

[3] min-you wu “on runtime parallel scheduling for processor load balancing” ieee transcations on parallel and distributed systems. Feb 1997(vol 8 no2) pp 173-186

[4] lm il,mull chong-wei xu,”A distributed drafting algoritham for load balancing” ieee transcations on software engineering October 1985(vol 11 no10) pp 1153--1161

[5] Jim Smith,paul Watson ,” DISTRIBUTED QUERY PROCESSINGON THE GRID”book Third International Workshop Baltimore, MD, USA, November 18, 2002 Proceedings , pp 279-290.

[6] Ehsan Saboori, Shahriar Mohammadi “A new scheduling algorithm for server farms load balancing” ieee truncations IIS , Volume: 1,Page(s): 417 – 420.

[7] Albert Y.Zomaya,yee-Hwei Tech “Observations on using gentic algorithms for dynamic load balancing” ” ieee transcations on parallel and distributed systems Sep 2001(vol 12 no9)pp 899-911.

[8] F bonomi,A. kumar “adaptive optimal load balancing in a nonhomogeneous multiserver system with a central job scheduler” ieee transcations on computer science.oct2000(vol 39 no 10) pp 1232-1250.

[9] Stanley Y.W. Su,Sanjay Ranka,Xiang He “Performance Analysis of Parallel Query Processing Algorithms for Object-Oriented Databases” ieee transcations on knowledge and data engineering November/December 2000(vol.12 no.6) pp 979-997.

[10] C.C. Chang,M. Templeton,D. Brill,E. Lund “Query Processing in a Fragmented Relational Distributed System” IEEE transactions on software engineering august 1985(vol 11 no.8) pp 795-810.

[11] null To-Yat Cheung “A Method for Equijoin Queries in Distributed Relational Databases” IEEE Transcations on computers august 1982(vol 31 no 8) pp 746-751.

[12] L. Harada, N. Akaboshi, M. Nakano, “An Effective Parallel Processing of Multiway Joins by Considering Resources Consumption,” Proc. Int'l Conf. Computing Information, 1994.

[13] M.S. Chen,P.S. Yu “Interleaving a Join Sequence with Semijoins in Distributed Query Processing” IEEE transcations on parallel and distributed processing ,sep-1992(vol 3 no 5) pp 611-621

[14] Klaithem Al,Nuaimi,Nader Mohamed,Mariam Al Nuaimi,Jameela Al-Jaroodi “A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms” Network Cloud Computing and Applications (NCCA) dec-2012 pp 137-142

[15] Laercio L. Pilla,Philippe O.A. Navaux,Christiane P. Ribeiro,Pierre Coucheney,Francois Broquedis,Bruno Gaujal,Jean-Francois Mehaut “Asymptotically Optimal Load Balancing for Hierarchical Multi-Core Systems” IEEE transactions Computer Science Dec-2012 pp 236-243.

[16] Harshitha Menon,Nikhil Jain,Gengbin Zheng,Laxmikant Kale “Automated Load Balancing Invocation Based on Application

Characteristics" International Conference on Cluster Computing (CLUSTER) sep 2012 pp 373-381

[17] Anubhav Sharma, Om Prakash Pal and co "An optimal approach of load balancing for grid computing" Proceedings of the CUBE International Information Technology Conference iisue date sep-2012

-567-571

[18] Klaithem Al Nuaimi,Nader Mohamed "A Survey of Load Balancing in Cloud Computing Challenges and Algorithms" Publish on Network Cloud Computing and Applications (NCCA) December 2012 pp-137-142

[19] Jasma Balasangameshwara,Nedunchezhian Raju "Performance-Driven Load Balancing with a Primary-Backup Approach for Computational Grids with Low Communication Cost and Replication Cost" IEEE Transactions on Computers May 2013 pp 990-1003

[20] Carlos Oliveira "Load Balancing on Virtualized Web Servers" publish on 41st International Conference on Parallel Processing Workshops (ICPPW) sep-2012 pp-340-347

[21] Yang Jiao, Wei Wang "Design and Implementation of Load Balancing of Distributed-system-based Web Server", Electronic Commerce and Security, International Symposium, issues date july 2010

pp-337-342

[22] Frode Eika Sandnes, Yo-Ping Huang "Temporal Load-Balancing of Web-Server Traffic" published on International Conference on Parallel and Distributed Computing Applications and Technologies, December 2006 ,pp. 266-271

[23] Rina Su, Yu Wang “Immune Genetic Algorithm-based Load Balancing in Web Cluster” ieee International Conference on Genetic and Evolutionary Computing ,December 2010,pp 321-324

[24] Sudsanguan Ngamsuriyaroj, Pak Rattidham, Ittipon Rassameeroj, Pisut Wongbuchasin, Nuttida Aramkul, Suttatip Rungmano “Performance Evaluation of Load Balanced Web Proxies” IEEE International Conference on Advanced Information Networking and Applications Workshops, March 2011,pp. 746-750.

[25] Zhaolei Duan, Zhimin Gu ,” Dynamic Load Balancing in Web Cache Cluster” IEEE International Conference on Grid and Cloud Computing , October 2008pp. 147-150.

[26] Wu Yang, ShuangQing Li, DaiJie Cheng” A Load Balancing Strategy in Web Cluster System “IEEE International Conference on Natural Computation August 2007 pp. 809-813.

[27] Rocco Aversa, Beniamino Di Martino “Load Balancing of Mobile Agents Based Applications in Grid Systems” IEEE International Workshops on Enabling Technologies . June 2008 pp. 6-11.

[28] Hamid Mcheick,Ziad Rajih Mohammed,Abbass Lakiss “Evaluation of Load Balance Algorithms” ACIS International Conference on Software Engineering Research, Management and Applications Issue Date:August 2011 pp. 104-109.

[29] Yinfei Pan, Wei Lu, Ying Zhang, Kenneth Chiu “A Static Load-Balancing Scheme for Parallel XML Parsing on Multicore CPU” IEEE International Symposium on Cluster Computing and the Grid. Issue Date:May 2007,pp. 351-362

[30] Weiguang Shi, Lukas Kencl “Sequence-preserving adaptive load balancers” Symposium On Architecture For Networking And Communications Systems” Issue Date:December 2006,pp. 143-152

University of Bedfordshire

7.1 Appendix 1:

7.1.0 Poster

LOAD BALANCING IN DISTRIBUTED QUERY MANAGEMENT AT WEB ENABLED SYSTEM

University of Bedfordshire

Satya Vadapalli(1126828)
Dr WEI HUANG

MSc Computer Science

Introduction

Today digital world every business turn to internet market to establish its mark, So usage of web application are increased exponentially. The these application are multi tire applications, for them replication strategy is Must . To adjust the load among the severs are performed by load balancers. These load balancers are often considered as Dispatchers.

Problem Statement

Ever increase in web based application often leads to load problems and application scaling issues at Servers. Existing load balancing techniques does not having full control of the application at Clients browser end.

Aim & Objectives

Design a load balancer with dynamically adjust load among the servers in Distributed web based systems.

- Proposed Load balancer is dynamically adjust the clients requests among the servers.
- Takes clients side weight into consideration.
- Reduce query processing time.
- Increase overall efficiency of the web based applications

Contextual Review

In Literature we have frame works available to design the distributed system that are COBRA ,RM-ODP,DCE and HADOOP.

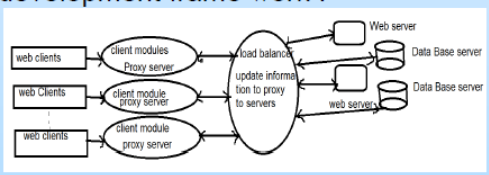
First come first serve(FCFS),Round Robin(RB),weighted Round Robin(WRB), Supervised learning algorithm Back propagation from artificial neural networks(ANN)

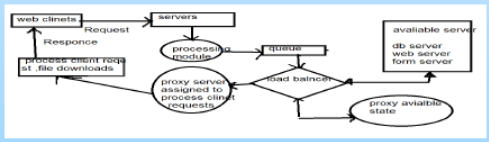
Proposed methodology

Our proposed methodology to design an heuristic load balancer uses both combination of weighted round robin technique and back propagation techniques.

Design & Development

The proposed system was designed and developed by using agile software development frame work .

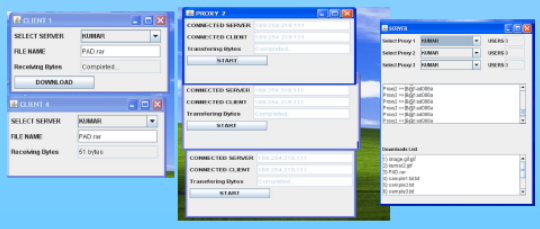




The above figure represents design and development prototype data flow diagrams

Execution and Testing

The developed system was tested as per our considered input and we successfully produced expected output.



Conclusion

Our proposed heuristic Load balancer for Distributed query management for web enabled systems shows better results than other load balancers

7.1.1 Project Proposal Form

FACULTY OF CREATIVE ARTS, TECHNOLOGIES AND SCIENCE

Form for Research Ethics Projects (CATSethicsform)

1.	Student Name	SATYA VADAPALLI
2.	Student Number:	1126828
3.	Degree Pathway:	MSc COMPUTER SCIENCE
4.	Supervisor's name	Dr Wei Huang
5.	Supervisor Signature	
6.	Working title of project	Load balancing in distributed query management at web enabled systems

SECTION A Proposal

Please summarise below the ethical issues involved in the research proposal and how they will be addressed. In any proposal involving human participants clear explanation of how informed consent will be obtained, how confidentiality will be observed, how the nature of the research and the means of dissemination of the outcomes will be communicated to participants must be provided.

Brief outline of project

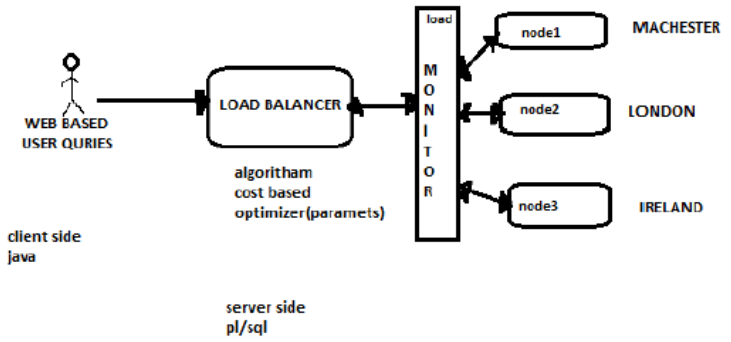
List of ethical issues

University of Bedfordshire

MSc Project Proposal Form

AY12/13, Semester 1

Student Number	1126828
Student Name	Satya vadapalli
Degree Course	MSc Computer Science
Supervisor Name	Dr Wei Huang
Title of Project	Load balancing in distributed query management at web enabled systems
Description of your artefact	Today internet is growing rapidly with peta byte rate, so information exists there is enormous and needs distributed approach to access those data. Load balancing in these environments plays a crucial role to access information from distributing systems. Existing System: In general cases load balancers are sitting at the middle of user systems and servers then calculate the load at server level then pass the input queries to small load node, in this way it will evenly distribute the load among the servers. I am proposing heuristic based load balancer to calculate the load on the system with other cost based parameters at run time in dynamic environments. In proposed system load balancer get the additional parameters from client side and rest of the information from server side then adjust the load among the nodes, and prioritize the load among the servers with business logic i.e less or optimized cost. The following diagram gives general outlook of the proposed prototype.

	 In proposed prototype uses client –server model ,two modules resides one at load balancer another at server side , server side module get the statistics from nodes and pass to load balancer ,in load balancer it have input requests from clients it uses both information and take intelligent decision which request goes to which node ,to make that decision it uses rank based approach with circular fashion and estimate the load by using back propagation technique .
What methodology (structured process) will you be following to realise your artefact?	I am using agile methodology to produce the proposed prototype. The steps involved requirement analysis, design a prototype, then develop a load balancer, use client systems for input load generation and use two are three nodes to test the proposed scenario and tabulate the results.
How does your project relate to your degree course and build upon the units/knowledge you have studied/acquired	The proposed thesis related to MSc computer science subjects mainly system architecture and online databases , artificial intelligence and knowledge from rest of the modules.
Resources	Open source softwares (mysql) and trail software oracle 10g,11g and JDK(java development kit) Vmware workstation for virtual environment to simulate the proposed prototype.
Have you completed & submitted your ethics form?	Yes

SECTION B Check List

Please answer the following questions by circling **YES** or **NO** as appropriate.

1. Does the study involve vulnerable participants or those unable to give informed consent (e.g. children, people with learning disabilities, your own students)?

   YES **NO** ✓

2. Will the study require permission of a gatekeeper for access to participants (e.g. schools, self-help groups, residential homes)?

   YES **NO** ✓

3. Will it be necessary for participants to be involved without consent (e.g. covert observation in non-public places)?

   YES **NO** ✓

4. Will the study involve sensitive topics (e.g. obtaining information about sexual activity, substance abuse)?

   YES **NO** ✓

5. Will blood, tissue samples or any other substances be taken from participants?

   YES **NO** ✓

6. Will the research involve intrusive interventions (e.g. the administration of drugs, hypnosis, physical exercise)?

   YES **NO** ✓

7. Will financial or other inducements be offered to participants (except reasonable expenses or small tokens of appreciation)?

   YES **NO** ✓

8. Will the research investigate any aspect of illegal activity (e.g. drugs, crime, underage alcohol consumption or sexual activity)?

   YES **NO** ✓

9. Will participants be stressed beyond what is considered normal for them?

   YES **NO** ✓

10. Will the study involve participants from the NHS (patients or staff) or will data be obtained from NHS premises?

    YES **NO** ✓

If the answer to any of the questions above is "Yes", or if there are any other significant ethical issues, then further ethical consideration is required. Please document carefully how these issues will be addressed.

Signed (student): V. Satyakumar Date: 20/3/2013.
Countersigned (Supervisor): Date: